

## H-4. loadmods, fetchmods

### 1.0 General Information

#### 1.1 Application Description

The loadmods program reads the mod files used by NWSRFS and inserts each mod into the proper database table. The fetchmods program retrieves mods from the database and writes them to a file in the proper format. The loadmods program empties out the mod file as it stores mods to the database and the fetchmods program repopulates the mod files based on a user specified time period. It is strongly recommended that the user copy the mod files from their operational location to the default mods archive directory  $(\$(adb\_dir)/data/nwsrfs/mods)$  before running these programs. Attachment A lists all the mods that have been set up for database storage and retrieval. These programs are written in ecpg/C.

#### 1.2 Design Considerations

Each mod has its own database table that is based on one of five mod formats (see NWSRFS User Documentation Section VI.5.3C **and Attachment A**) and then customized to meet individual needs. Table names are *modmodname* (i.e. *modsetqmean*). Simplified procedural flowcharts are provided in Attachment B.

#### 1.3 Application Assumptions

The loadmods program assumes all dates and times to be in Z time, if another time zone code is specified it will not parse the mod and returns an error. It handles date formats as described in NWSRFS User Documentation Section VI.5.2B dated 8/13/98.

#### 1.4 Enhancements/Bug Fixes/Changes

##### **Build OB7.2**

This application was updated to take into account the upgrade of the OS in ob7.2 and the change in RDBMS from Informix to Postgres. Documentation was updated to reflect this change.

### 2.0 Configuration Information

These programs use the following apps\_defaults tokens:

adb_name	archive database name
adb_dir	archive base directory

These programs make use of the following database tables (which must be populated):

*fgroup*  
*fgroupseg*  
*modctrl*

The *fgroup* and *fgroupseg* tables can be populated using the *group\_parse* program (see Section H-3).

The *modctrl* table is already populated when the archive database is delivered. It consists of four columns: “mod\_name”, “load”, “fetch\_oper”, and “fetch\_spin”. The “load” and “fetch” columns are simple on/off switches with 1=on and 0=off. The “fetch\_oper” column refers to real-time (operational) runs, while the “fetch\_spin” column refers to historical runs. All mods are initially turned ‘on’ when the archive database is first installed. Each site may then customize it as they wish.

Input and output mod file locations are described in the User How-To section.

Log files for both programs are written to  $\$(adb\_dir)/logs/nwsrfs/mods$ .

### 3.0 User How-To

Both of these programs are run on the command line by simply typing the program name and the required inputs. Both programs also allow for optional command line arguments that control the ‘test’ and ‘debug’ modes (these must be entered before any other arguments if they are used):

-t = test mode; do not write to database  
-d = debug mode; write everything to the log file

#### **loadmods**

Loadmods has three command line arguments:

1. filename or carryover group indicator (-f or -c)
2. filename or carryover group id
3. directory to read mod files from if different from  $\$(adb\_dir)/data/nwsrfs/mods$  (to be known as *mdir* throughout the rest of this documentation) - optional

When the -f (filename) option is used, the program will simply parse the mods in the file indicated. When the -c (carryover group) option is used, the program assumes the filenames to be the same as the forecast group ids found in the *fgroup* table by searching on the carryover group id provided. It will look for the files in the directory provided on the command line or, if no path is provided, in the *mdir*. While the user can

point the application to a directory other than the default location ( $\$(adb\_dir)/data/nwsrfs/mods$ ), such as the operational location of the mods files, it is not recommended. The user should copy mod files from their operational location to the default location.

The program uses the *modctrl* table to determine which mods should be parsed and stored in the database.

Once the mods are parsed and written to the database they are deleted from the mod file. Mods which could not be parsed due to incorrect format, or not stored because they are not defined within the program or because of settings in the *modctrl* table, are kept in the mod file, along with any comment lines.

### **fetchmods**

Fetchmods has seven command line arguments:

1. mode ('oper' or 'spin')
2. LSTCMPDY (ccyymmddhh) – in Z time
3. number of days back (from LSTCMPDY) to start of run
4. number of days forward (from LSTCMPDY) to end of run
5. forecast or carryover group indicator (-f or -c)
6. forecast or carryover group id
7. directory to write mod files to if different from mdir - optional

The oper/spin mode option simply tells the program which column of the *modctrl* table to read to determine which mods to write into the mod files. The 'oper' mode is for operational (real-time) model runs and usually uses all mods available. The 'spin' mode is for historical (past event) model runs and may be more selective in which mods are included.

When the -f (forecast group) option is used, the program searches the *fgroupseg* table to determine which segments belong in the mod file for that group. When the -c (carryover group) option is used, the program first searches the *fgroup* table to find the forecast groups and then uses the *fgroupseg* table to find the segments. The mod files use the same name as the forecast group id and are written to the directory provided on the command line or, if no path is provided, in the mdir.

Fetchmods appends the retrieved mods to whatever is already in the mod files (files are not overwritten).

## 4.0 Troubleshooting Information

Check the log files to see if mods are being parsed correctly. Mods that are not being stored to the database most likely are either turned 'off' in the *modctrl* table or have an incorrect format. If problem is not resolved, contact the RFC Support Group.

## 5.0 References

NWSRFS Users Manual Documentation Section VI.5.2B  
NWSRFS Users Manual Documentation Section VI.5.3C  
Archive Database data dictionary

## Attachment A Defined mods

### FORMAT A1

- SETMSNG

### FORMAT A2

- TSADD
- TSCHNG
- TSMULT
- TSREPL

### FORMAT B1

- RAINSNOW

### FORMAT B2

- APICCO
- IGNORETS
- SACCO
- SWITCHTS
- UHGADJ
- XINCO
- SSARREG (special case B2a within programs)

### FORMAT B3

- AEICQN
- AESCCHNG
- AIADJ
- APICBASF
- BUBLSHFT
- CHGBLEND
- MATCHNG
- MFC
- QCSHIFT
- QPSHIFT
- ROCHNG
- ROMULT
- RRICNG
- RRIMULT
- SACBASEF
- SETQMEAN
- UADJ
- UHGCHNG
- WEADD
- WECHNG
- ZERODIFF

## Attachment B Flow Charts

